**Session bean in java**

**What is session bean?**

Session bean are reusable component that contain business logic. Conversation with client involves interaction between client and bean which is composed of number of method calls. Session bean represent work performed by single user and hold state between method calls. There must be separate instance of bean for each user. Session bean are nonpersistent which is not stored to permanent storage. As everything on server have lifecycle session bean also have lifecycle which is maintained by server.

**How session bean differ from other bean type?**

Session bean is different from other bean type in term of scope of their lifetime as it’s instance is comparatively short-lived object . Session bean have lifetime equivalent of session or client calling instance of session bean. The EJB container is responsible for creating instance of session bean and later when client disconnects EJB container is empowered to destroy session bean instance. Entity bean remain in memory for months or year. In contrast if client is using session bean for 20 minutes then it might be in memory for minutes to hours, but probably not week or months.

**Advantage of session bean**

**Disadvantage of session bean**

**Type of session bean**

1. **Stateless session bean**
2. **Stateful session bean**
3. **Singleton session bean**
4. **Stateless bean**

Stateless bean is used in single method invocation where we don’t need to retain state from method to method. After each method call EJB container may destroy it, or clear all information related to past method invocation. Example of stateless session bean is registration form where session bean take information of one user and when another user come session bean don’t maintain information of previous request. Stateless session bean could be reused and swapped from one client to another on each method call.

**Lifecycle of stateless bean**

**Bean instance does not exist**

When nobody is demanding session bean there does not exist any session been instances. So instance will only appear when there is client. When someone approach to site container decide whether to create new instance or not. Now here container decide whether to create new instance or not on basic of number of user. If user goes beyond limit then server automatically create new instance, it is decided by server, it is not depended on our request. If number of client increase server automatically create new instances. In stateless session bean we might have 5 instance for 5000 user which is far better than 5000 instance for 5000 user. This 5 instance in memory called instance pool.

Once new instance is created it will inject any dependency injection if needed. So here sources on which EJP is depended can be called.

Stateless have four callback

1. post construct

Post construct callback is called as soon as instance is created. Once post construct is called there is pool of equivalent business method which is instantiated which you can easily invoke.

1. pre destroy

pre destroy is called in situation where project is deployed, here you can free resources.

1. init
2. remove

When there is no client in that case server is going to wait for some time it may be hour, two hour or by default one week. If in one week no client comes then server think that this EJB is not required and that EJB bean is destroy.

So we can destroy bean in two way, First is that we deploy EJB and second is server decide. After that if client comes then new instance is created and lifecycle start again.

1. **Stateful bean**

Stateful session bean is used to retain state of individual client. During method invocation if state of method change then client get update state in next method invocation. Stateful session bean is designed for conversation which include multiple method invocation. We must use stateful session bean in application where user is updating any list. For example user is selecting number of channels user want to watch in TV.

**Lifecycle of stateful bean**

In stateful bean number of instance is equal to number of clients which means if 5000 clients connect to server then EJB will create 5000 instances of stateful bean. This is big problem that is solved by passivation and activation. In stateful bean all client will be connected but not all of them will call method all time. Consider while 5000 client is connected might be 100 client is continuously calling method. In this case server passivate remaining 4900 client and state of their method are stored in virtual memory or secondary storage. If any passivate client request for method then it is activated.

1. Singleton session bean

**Methods of session bean**

1. ejbCreate()
2. ejbPassivate()
3. ejbActivate()
4. ejbRemove()

1)ejbCreate()

Programmer can perform any initialization task in this method for example setting variable value. Programmer have to provide at least one ejbCreate() method but programmer can define multiple ejbCreate() method with different arguments.

2)ejbPassivate()

This method is called immediately before session bean is passivated. In this method we can release any resources hold by bean. ejbPassivate() is warning to bean that it’s state is being about to swapped out.

3)ejbActivate()

This method is called immediately before session bean is activated. In this method we can acquire resources that we have released during ejbPassivate().

4)ejbRemove()

ejbRemove() is clean-up method. This method is called by server immediately before bean is removed from memory and we can free all resources that we have allocated to bean.